Query-Drift Prevention for Robust Query Expansion

Kunal Suthar
Dhirubhai Ambani Institute of Information and Communication Technology
Gandhinagar, Gujarat, India
201401131@daiict.ac.in

ABSTRACT

The automatic query expansion retrieval based on pseudo-relevance feedback(also known as the blind-relevance feedback) performs effectively on average, but the performance is inferior to that of using the original query for many information needs. Further, the average precision(MAP) score for a query, dwindles on using the query expansion based on pseudo-relevance feedback. The important cause for this robustness decrease(the plummeting MAP score for a query), that is, the query drift problem, would be attempted to be solved. The method used to ameliorate the query drift problem, is by the fusion of the results retrieved in response to the original query and to its expanded form[1]. This fusion-based approach produces a retrieval performance, which is better than that of retrieval based only on the original query and a more robust performance(the improvement in the average precision per query) than that of retrieval using the expanded query.

Keywords

query expansion, pseudo feedback, robust query expansion, fusion, query drift

1. INTRODUCTION

A pseudo-relevance-feedback based query expansion method upgrades a query with terms from the top k highly ranked documents by an initial search [2]. While this method produces effective performance on average, the performance is sometimes inferior to that of the original retrieval [3,4,5]. This depletion in performance is because of the terms which were augmented to the original query using the top k documents. Now, it is possible that the top-k documents retrieved by the original query were not relevant and were not required by the user. As we are using pseudo-relevance-feedback, it would blindly augment the query based on the top weighted terms from the top-k ranked documents and those documents in-turn were non-relevant. So, we are upgrading the query based on the documents which are non-relevant, which is likely to decrease the robustness of a query. This robustness problem is known as the query-drift problem[6]:the shift in intention from the original query to its expanded form. The prime motivation which lies behind the approach in query drift prevention is that the documents ranked high by both retrieved lists are potentially relevant since they constitute a good match to both forms of the presumed information need. The approach used for query drift prevention is the fusion method, that is, we fuse the ranked lists of the initial retrieval and the ranked list post pseudo-relevance feedback, based on a hybrid score[1].

2. RETRIEVAL METHODS

Assume q, d, and $Score_{init}(d|q)$ to denote a query, a document,and a score assigned to d in response to q by the initial search, respectively; D_{init} denotes the list of documents most highly ranked according to $Score_{init}(d|q)$. We assume that some pseudo-relevance-feedback-based query expansion approach uses information from some documents in D_{init} for ranking the entire corpus and that PF(D_{init}) is the resultant list of highest ranked documents; $Score_{pf}(d|q)$ denotes the score assigned to d by the pseudo-relevance-feedback-based retrieval.

2.1 Algorithms Implemented

The following retrieval methods essentially operate on Dinit U PF(Dinit).

The **combMNZ** method [1] rewards documents that are ranked high in both Dinit and PF(Dinit):

$$Score_{combMNZ}(d|q) \stackrel{def}{=} (\delta[d \in \mathcal{D}_{\text{init}}] + \delta[d \in PF(\mathcal{D}_{\text{init}})]) \cdot \left(\frac{\delta[d \in \mathcal{D}_{\text{init}}]\, Score_{\text{init}}(d|q)}{\sum_{d' \in \mathcal{D}_{\text{init}}} Score_{\text{init}}(d'|q)} + \frac{\delta[d \in PF(\mathcal{D}_{\text{init}})]\, Score_{\text{pf}}(d|q)}{\sum_{d' \in PF(\mathcal{D}_{\text{init}})} Score_{\text{pf}}(d'|q)}\right).$$

The **interpolation** algorithm[1] differentially weights the initial score and the pseudo-feedback-based score using an interpolation parameter :

$$Score_{interpolation}(d|q) \stackrel{def}{=} \frac{\lambda\delta[d \in \mathcal{D}_{\text{init}}]\, Score_{\text{init}}(d|q)}{\sum_{d' \in \mathcal{D}_{\text{init}}} Score_{\text{init}}(d'|q)} + \frac{(1-\lambda)\delta[d \in PF(\mathcal{D}_{\text{init}})]\, Score_{\text{pf}}(d|q)}{\sum_{d' \in PF(\mathcal{D}_{\text{init}})} Score_{\text{pf}}(d'|q)}.$$

The **re-rank** method[1], re-orders the (top) pseudo-feedback based retrieval results by the initial scores of documents:

$$Score_{re-rank}(d|q) \stackrel{def}{=} \delta[d \in PF(\mathcal{D}_{\text{init}})]\, Score_{\text{init}}(d|q).$$

.

3. EVALUATION METRICS

The evaluation methods used are:

1. **MAP(Mean Average Precision)**: The standard retrieval evaluation score which takes the arithmetic mean of all the values of MAP at k, is used to calculate the $Score_{init}(d|q)$ and $Score_{PF}(d|q)$, based on which the fusion based scores are computed for every document-query pair.

2. **<Init(%)**: Percentage of queries for which the expansion-based MAP performance is worse than that of using the original query(measure of robustness).

4. IMPLEMENTATION

4.1 Corpus used

Corpus	Queries	Disks
WSJ	151-200	1-2
SJMN	51-150	3
AP	51-150	1-3
ZF	51-200	1-3

The corpus used in the experiment were from the TREC1-3 disks, which incorporated the above data-sets WSJ, SJMN, AP and ZF. The title of the topics were taken as the queries and the queries which applied for the previously mentioned data-sets are 151-200, 51-150, 51-150 and 51-200 respectively.

4.2 Experimental Setup

For conducting the experiments, the TREC corpora, stated above, was used, with the *topics* titles serving as the queries. The generation of the ranked lists for the initial retrieval and the pseudo-relevance-feedback retrieval were done on the Terrier search engine version 3.6. The initial retrieval weighting model used to score documents for a query in terrier, is the InexpC2 model[7], as it was giving the best results amongst the weighing models available in terrier. The D_{init} was set to the 1000 documents with the highest initial ranking score $Score_{init}(d|q)$. The set PF(D_{init}) of 1000 documents was generated by using the pseudo-relevance-feedback query expansion functionality of terrier which uses the default query expansion model called Bo1. The IDE used for conducting the experiments on terrier was Eclipse Luna, as terrier is written in Java.

The first step involves specifying the location of the queries and the qrels, choosing the desired weighting models in the terrier.properties file, and then indexing the corpus. After the indexing is complete, we run the retrieval on the queries. After the retrieval is done, we get an initial ranked list(D_{init} of 1000 documents) with the scores for every query-document pair. Then, we run the query expanded retrieval and get the new ranked list PF(D_{init}). After getting both the lists, the fusion-based algorithms are run on the D_{init} U PF(D_{init}). The fusion-based scores($Score_{combMNZ}(d|q)$, $Score_{interpol}$(d|q), $Score_{rerank}(d|q)$) are calculated for every query-document pair and the results are stored in a hashmap for every query. The key and values in the hashmap are then transferred into a treemap which would keep all the documents in sorted order based on the fused-score, that is the value in the hashmap with the key being the document number. The new ranked list is generated on the basis of the sorted scores in the treemap and the evaluation is done on its basis now. The MAP scores and <Init %(percentage of queries(denoted by <Init %) for which the MAP performance is worse than that of the initial ranking. Its lower value corresponds to improved robustness.) are evaluated on the new ranked list, using functionalities of terrier.The results are then compared with the paper[1].

5. EXPERIMENTAL RESULTS

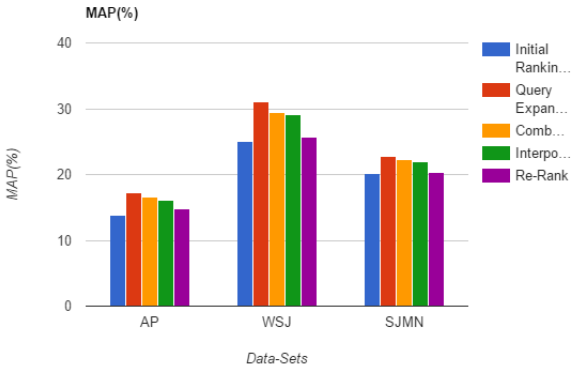

Figure 1: Retrieval performance(MAP%) on different algorithms

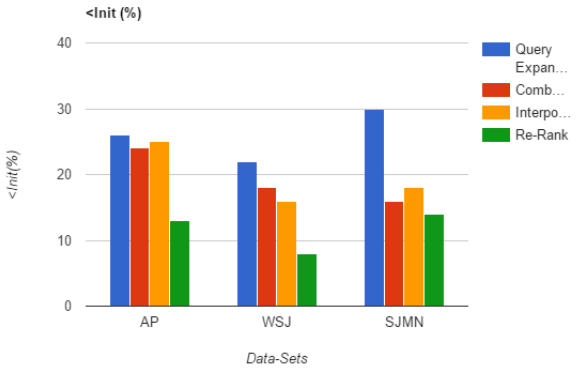

Figure 2: Robustness on different algorithms

In the above figure 1, we observe that all fusion-based methods gives MAP performance, better than the MAP score of the initial ranking, that employs only the original query. The MAP scores are different from the conducted experiments and the scores in [1], but the variation in the scores resonates, that is the fusion-based results give better MAP performance than the initial retrieval and it also gives more robust results than the one using pseudo relevance based query-expansion(shown in figure 2). The combMNZ

Table 1: MAP and Robustness performance on data-sets AP and WSJ

	AP DATASET				WSJ DATASET			
ALGORITHM	MAP(%)	MAP(From paper[1])	<Init(%)	<Init(%)(From paper[1])	MAP(%)	MAP(From paper[1])	<Init(%)	<Init(%)(From paper[1])
Initial Ranking	13.89	22.2	-	-	25.00	27.8	-	-
Terrier's Bo1 Model(Expanded Model)	**17.25**	28.5(RM-1) **29.1(RM-3)**	26	38.4(RM-1) 28.3(RM-3)	**31.00**	33.2(RM-1) **34.7(RM-3)**	22	34.0(RM-1) 28.0(RM-3)
combMNZ(Fusion Method)	16.51	26.9	24	21.2	29.47	31.1	18	**14.0**
Interpolation(Fusion Method) at lambda=0.6	16.08	28.6	25	31.3	29.06	34.0	16	26.0
Re-Rank(Fusion Method)	14.79	25.9	**13**	**20.2**	25.65	29.8	**8**	22.0

Table 2: MAP and Robustness performance on data-sets SJMN and ZF

	SJMN DATASET				ZF DATASET	
ALGORITHM	MAP(%)	MAP(From paper[1])	<Init(%)	<Init(%)(From paper[1])	MAP(%)	<Init(%)
Initial Ranking	20.19	18.9	-	-	5.13	-
Terrier's Bo1 Model(Expanded Model)	**22.76**	24.1(RM-1) **24.6(RM-3)**	30	37(RM-1) 29(RM-3)	5.59	22
combMNZ(Fusion Method)	22.28	21.6	16	20	5.7	20
Interpolation(Fusion Method) at lambda=0.6	22.01	23.6	18	27	**6.5**	16
Re-Rank(Fusion Method)	20.37	20.4	**14.002**	**16**	5.99	**14.002**

and interpolation methods give MAP performance that is never worse to a significant degree than that of the expanded model. We also observe that the re-ranking algorithm yields the most robust performance(lowest < Init %).Table 1 and 2 above, shows the complete results.

6. CONCLUSION

The results generated by the experiments were in accordance with [1]. The fusion of the lists and generating a new ranked list based on the response to a query and its expanded form gives us a significant retrieval performance than based on the query alone. This fusion-based performance is also consistently more robust than that of the performance on the expanded query form.

7. ACKNOWLEDGMENTS

Sincere thanks to Prof. Prasenjit Majumder for his valuable guidance and suggestions for carrying out the experiments related to the approach. Also, thanks to Mr. Parth Mehta for providing the appropriate TREC datasets, the necessary queries and qrels related to it, required for working on this task.

8. REFERENCES

[1] L.Zighelnic and O.Kurland. Query-Drift Prevention for Robust Query Expansion: SIGIR08, July 20 to 24, 2008, Singapore.ACM 978-1-60558-164-4/08/07,pages 825826.

[2] C. Buckley, G. Salton, J. Allan, and A. Singhal. Automatic query expansion using SMART: TREC3. In Proceedings of TREC-3, pages 69 to 80, 1994.

[3] G. Amati, C. Carpineto, and G. Romano. Query difficulty, robustness, and selective application of query expansion. In Proceedings of ECIR, pages 127 to 137, 2004.

[4] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. A language modeling framework for selective query expansion. Technical Report IR-338, Center for Intelligent Information Retrieval, University of Massachusetts, 2004.

[5] K. Collins-Thompson and J. Callan. Estimation and use of uncertainty in pseudo-relevance feedback. In Proceedings of SIGIR, pages 303 to 310, 2007.

[6] M. Mitra, A. Singhal, and C. Buckley. Improving automatic query expansion. In Proceedings of SIGIR, pages 206 to 214, 1998.

[7] Configuring Retrieval in terrier. http://terrier.org/docs/v3.5/configure_retrieval.html